

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-094290
(43)Date of publication of application : 16.04.1993

(51)Int.Cl. G06F 9/06
G06F 9/06

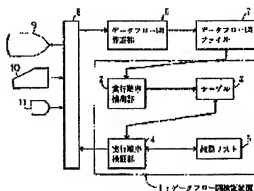
(21)Application number : 03-280800 (71)Applicant : SHIMADZU CORP
(22)Date of filing : 30.09.1991 (72)Inventor : INOUE FUJIO

(54) DATA FLOW CHART VERIFYING DEVICE

(57)Abstract:

PURPOSE: To automatically verify the description error of an executing order by chasing the executing order of processes set to a table by developing it to a route list, and executing error display by judging the circulation of the executing order when the same process appears.

CONSTITUTION: A monitor display 9 used for preparing a data flow chart, keyboard 10 and mouse 11 or the like are connected through an interface 8 to a data flow chart preparation part 6, and the data flow chart prepared at the data flow chart preparation part 6 is stored in a data flow chart file 7. In this case, a data flow chart verifying device 1 extracts the executing order relation of processes from the data flow chart in the data flow chart file 7 by using an executing order extraction part 2. The extracted executing order relation is registered on a table 3, and the route of data flowing between processes is developed from the contents to a route list 5 on a memory. During the process, it is verified whether the data flow chart can be successively executed or not and when it can not be executed, error output is executed.



(51)Int.Cl.¹

G 0 6 F 9/06

識別記号 庁内整理番号

4 3 0 E 8944-5B

4 4 0 U 8944-5B

F I

技術表示箇所

審査請求 未請求 請求項の数1(全 5 頁)

(21)出願番号 特願平3-280800

(22)出願日 平成3年(1991)9月30日

(71)出願人 000001993

株式会社島津製作所

京都府京都市中京区西ノ京桑原町1番地

(72)発明者 井上 藤男

京都市中京区西ノ京桑原町1番地 株式会社

島津製作所三条工場内

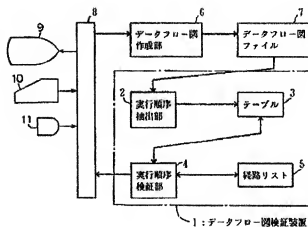
(74)代理人 弁理士 杉谷 勉

(54)【発明の名称】 データフロー図検証装置

(57)【要約】

【目的】 データフロー図の実行順序の記述誤りを自動的に検証する。

【構成】 データフロー図に記述されているプロセスの実行順序をデータの流れに基づいて抽出しテーブル3にセットする。テーブル3にセットされたプロセスの実行順序を経路リスト5に展開することで追跡し、その過程で同じプロセスが現れると実行順序が循環していると判断し、モニタディスプレイ9にエラー表示を行う。



【特許請求の範囲】

【請求項1】 情報処理システムにおける処理を機能単位ごとに分割し、各処理（プロセス）間でのデータの流れを図式化したデータフロー図の実行順序の記述誤りを検証する装置であって、データの流れに基づく各プロセスの実行順序を抽出する手段と、前記抽出されたプロセスの実行順序を追跡してデータの流れがプロセスの間で循環しているかを判断する手段と、前記データの流れの循環が判断されたときに外部に対してエラー出力を行う手段とを備えたことを特徴とするデータフロー図検証装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 この発明は、情報処理システムを分析、設計する際に作成されるデータフロー図の記述誤りを検証する装置に関する。

【0002】

【従来の技術】 データフロー図は、情報処理システムにおける処理を機能単位ごとに分割し、各処理（以下、プロセスという）間でのデータの流れを図式化したものである。データの流れだけに注目して作成されたものであるため、そのデータが何回ループ処理をうけるとか、どんな条件で分岐していくのか、というデータの制御構造を表記することはない。データの制御構造を表記するには、フローチャートのようなプロセスフロー図が別にある。ゆえに、データフロー図において、少なくともデータの流れがプロセスの間を循環することはなく、仮に、そのような記述であれば、データフロー図の実行順序に誤りがあると判断される。従来では、その実行順序の記述誤りの検証を人手に頼っている。

【0003】

【発明が解決しようとする課題】 したがって、記述誤りを発見しとすなどの検証ミスの発生を完全に避けることはできず、データフロー図が複雑化すればするほど、検証ミスの発生の確率は高くなる。また、検証を行う人の労力や時間も多大に費やされてしまう。

【0004】 この発明は、このような事情に鑑みてなされたものであって、自動的にデータフロー図の実行順序の記述誤りを検証できるデータフロー図検証装置を提供することを目的としている。

【0005】

【課題を解決するための手段】 この発明は、上記目的を達成するために次のような構成をとる。すなわち、この発明は、情報処理システムにおける処理を機能単位ごとに分割し、各処理（プロセス）間でのデータの流れを図式化したデータフロー図の実行順序の記述誤りを検証する装置であって、データの流れに基づく各プロセスの実行順序を抽出する手段と、前記抽出されたプロセスの実行順序を追跡してデータの流れがプロセスの間で循環しているかを判断する手段と、前記データの流れの循環が

判断されたときに外部に対してエラー出力を行う手段とを備えたことを特徴とする。

【0006】

【作用】 この発明の構成による作用は、次のとおりである。データフロー図のデータの流れから各プロセスの実行順序を抽出し、その実行順序を追跡することで、データの流れがプロセスの間を循環しているか否かを判断する。判断の結果、循環していれば、データフロー図の実行順序に記述誤りがあるとして外部にエラー出力を行う。

【0007】

【実施例】 以下、この発明の一実施例を図面に基づいて説明する。図1はこの実施例におけるデータフロー図検証装置の概略構成および、装置に取り付けられる周辺機器を示したブロック図である。データフロー図の作成に用いられるモニタディスプレイ9、キーボード10、マウス11などがインターフェイス8を介して、データフロー図作成部6に接続されている。データフロー図作成部6で作成されたデータフロー図はデータフロー図ファイル7に格納される。

【0008】 データフロー図検証装置1は、データフロー図ファイル7内のデータフロー図から、プロセスの実行順序関係を抽出する実行順序抽出部2と、抽出された実行順序関係を登録するテーブル3と、テーブル3の内容からプロセス間を流れるデータの経路をメモリ上の経路リストに展開し、その過程でデータフロー図が順次実行可能か否かを検証して、否の場合にインターフェイス8を介してモニタディスプレイにエラー出力を行う実行順序検証部4とを備えている。

【0009】 図2に検証の対象となるデータフロー図の一例を示す。九枠で囲まれた要素がプロセスを表しており、九枠内にはプロセス名が記述される。各プロセスを結合する矢印線がデータの流れ（以下、フローと称する）を示している。図では、プロセスP2→P3→P4→P2というようにフローがプロセスの間を循環しており、実行順序の記述誤りの例となっている。

【0010】 次に、図2のデータフロー図の実行順序の記述誤りを検証する動作について説明する。上記のように、データフロー図は特定の記述形式で作成されているので、実行順序抽出部2は、その記述形式にしたがってデータフロー図を解析する。すなわち、九枠で囲まれた図形要素をプロセスであると判断し、矢印線をフロー、矢印線の基部にある九枠の図形要素を接続元プロセス、矢印線の先にある九枠の図形要素を接続先プロセスであると判断する。

【0011】 この判断に基づいて、各フローの接続元プロセスと、接続先プロセスとを抽出し、図3に示すようなフォーマットのテーブル3にセットしていく。このとき、各プロセスを識別番号で表す。例えば、データフロー図の先頭にあるプロセスP1（このP1は実際には処理

3

名となっている)を番号「1」で表すと、以下のプロセスを全てその続き番号として表す。図2のフローAの接続元プロセスP1の識別番号を「1」とすると、その接続先にあるプロセスの識別番号を「2」で表すという具合である。

【0012】図2のデータフロー図から、図3に示すテーブル3が完成すると、実行順序検証部4が起動してデータフロー図の実行順序の記述誤り(フローが循環しているか否か)を検証する。その動作を図5のフローチャートに示して説明する。

【0013】まず、ステップS1で、テーブル3の先頭にあるフローAの接続元プロセスの識別番号「1」(以下、プロセス番号「1」と記す)と、接続先のプロセス番号「2」とを読み出し、図示しないメモリ上の経路リスト5に格納する。

【0014】ステップS2で、経路リスト5を読み込み、接続先のプロセス番号を読み出す。ここでは、プロセス番号「2」が読み出される。読み出したプロセス番号「2」をキーとして、テーブル3の接続元プロセスの欄にセットされているプロセス番号とのマッチングをとり、マッチングしたフローを読み出す(ステップS3)。プロセス番号「2」を接続元プロセスとしてもつフローBが読み出される。

【0015】ステップS4で、読み出したフローBの接続先のプロセス番号をテーブル3から読み出す。プロセス番号「3」が読み出されて、ステップS5に進み、そのプロセス番号「3」がすでに経路リスト5中に格納されているかどうかを判断する。格納されていなければ、ステップS6で、経路リスト5に格納して、ステップS2にリターンする。もし、経路リスト5内にすでに同じプロセス番号が格納されていれば、データフロー図の中にフローが循環している部分があるとし、ステップS7に進んで、モニタディスプレイ9にエラー表示を行う。

【0016】いま、経路リスト5には、ステップS1で格納したプロセス番号「1」とプロセス番号「2」しか格納されていないので、ステップS5の判断処理からステップS6に進んで、経路リスト5にプロセス番号「3」が格納される。そして、ステップS2に戻って、経路リスト5の接続先のプロセス番号、すなわち、「3」が読み出されて、ステップS3でプロセス番号「3」を接続元とするフローCが読み出される。次のス

4

テップS4でフローCの接続先のプロセス番号「4」が読み出されてステップS5からステップS6に進み、経路リスト5に格納される。ここまでの処理で、経路リスト5の内容は図4に示ようになる。

【0017】もう一度、ステップS2にリターンして上記の処理を繰り返すと、経路リスト5の接続先にあるプロセス番号「4」を元とするフローDの接続先のプロセス番号「2」が読み出される。プロセス番号「2」はすでに、図4の経路リスト5の中に存在しているからステップS5の判断処理でフローの循環が判明する。すなわち、フローの接続先のプロセスを追跡した結果、すでに通過したプロセスに帰着することになるため、フローの循環が判断され、ステップS7でエラーの表示が行われる。

【0018】

【発明の効果】以上の説明から明らかなように、この発明のデータフロー図検証装置は、プロセスの実行順序に基づいて、データの流れがプロセスの間を循環しているか否かを自動的に判断することにより、データフロー図の実行順序の記述誤りの検証を自動化した。したがって、人手による見落としなどの検証ミスがなく、また、検証を行う人の時間や労力を大幅に削減できる。データフロー図が複雑になればなるほどこの発明の効果は顕著に現れる。

【図面の簡単な説明】

【図1】この発明の一実施例に係るデータフロー図検証装置の概略構成を示したブロック図である。

【図2】検証の対象となるデータフロー図の一例である。

【図3】実施例におけるテーブルのフォーマットの一例を模式化した図である。

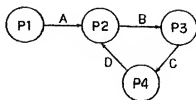
【図4】実施例における経路リストの一例を模式化した図である。

【図5】記述誤りの検証手順を示したフローチャートである。

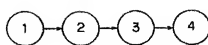
【符号の説明】

- 1・・・データフロー図検証装置
- 2・・・実行順序抽出部
- 3・・・テーブル
- 4・・・実行順序検証部
- 5・・・経路リスト

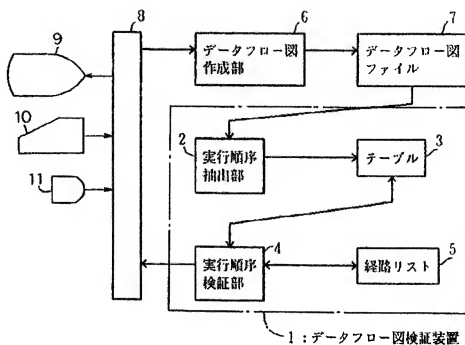
【図2】



【図4】



【図1】



【図3】

3: テーブル

フロー	接続元プロセス	接続先プロセス
A	1	2
B	2	3
C	3	4
D	4	2

【図5】

